Data Transfer Algorithm That Does Not Require High Latency Read Operations

ABSTRACT

A mechanism is provided for the controlled transfer of data across LDT and PCI buses without requiring any high latency read operations. The preferred embodiment of the invention removes the need for any read accesses to a remote processor's memory or device registers, while still permitting controlled data exchange. This approach provides significant performance improvement for systems that have write buffering capability. In operation, each processor maintains a set of four counters that are organized as two pairs, where one pair is used for the transmit channel and the other pair is used for the receive channel. At the start of an operation all counters are initialized to zero and are of such size that they cannot wrap, *e.g.* they are at least 64 bits in size. One processor, *e.g.* processor "B," allocates receive buffer space locally and transfers the addresses to another processor, *e.g.* processor "A." Processor "B" increments a "Local Rx Avail" counter by the number of local buffers and then writes this updated value to a "Remote Tx Avail" counter in processor A's memory. Processor "A" is now able to transfer data packets. It increments a "Local Tx Done" counter after each packet is sent until "Remote Tx Avail" minus "Local Tx Done" is equal to zero. At any time, the current value of the "Local Tx Done" counter on processor "A" can be written to the "Remote Rx Done" counter on processor "B". Processor "B" can determine the number of completed transfers by the subtraction of

"Remote Rx Done" from "Local Rx Avail" and can process these buffers accordingly. Once processed, the buffers can be freed or re-used with the cycle repeating when processor "B" again allocates receive buffer space locally and transfers the address to processor "A". The transmit channel from processor "B" to processor "A" is a mirror image of the procedure described above.